

FIG. 1

FIG. 2 is a block diagram of a broadcast receiver system 120. The system includes a user input receiver 224, a user input decoder 222, a microprocessor 210, an IR command encoder 226, an IR emitter 228, a tuner 202, a data extractor 206, a graphics overlay generator 216, a line driver 230, a communication port 232, a display 218, and three storage devices 212, 214, and 213. The system is connected to a bus 208. The microprocessor 210 is connected to the user input decoder 222, the user input receiver 224, the IR command encoder 226, the IR emitter 228, the tuner 202, the data extractor 206, the graphics overlay generator 216, the line driver 230, and the communication port 232. The tuner 202 is connected to the data extractor 206. The data extractor 206 is connected to the graphics overlay generator 216. The graphics overlay generator 216 is connected to the display 218. The line driver 230 is connected to the communication port 232. The communication port 232 is connected to the display 218. The display 218 is connected to the storage device 213. The storage device 212 is connected to the bus 208. The storage device 214 is connected to the bus 208. The storage device 213 is connected to the bus 208. The bus 208 is connected to the engine 217 and the operating system 219.

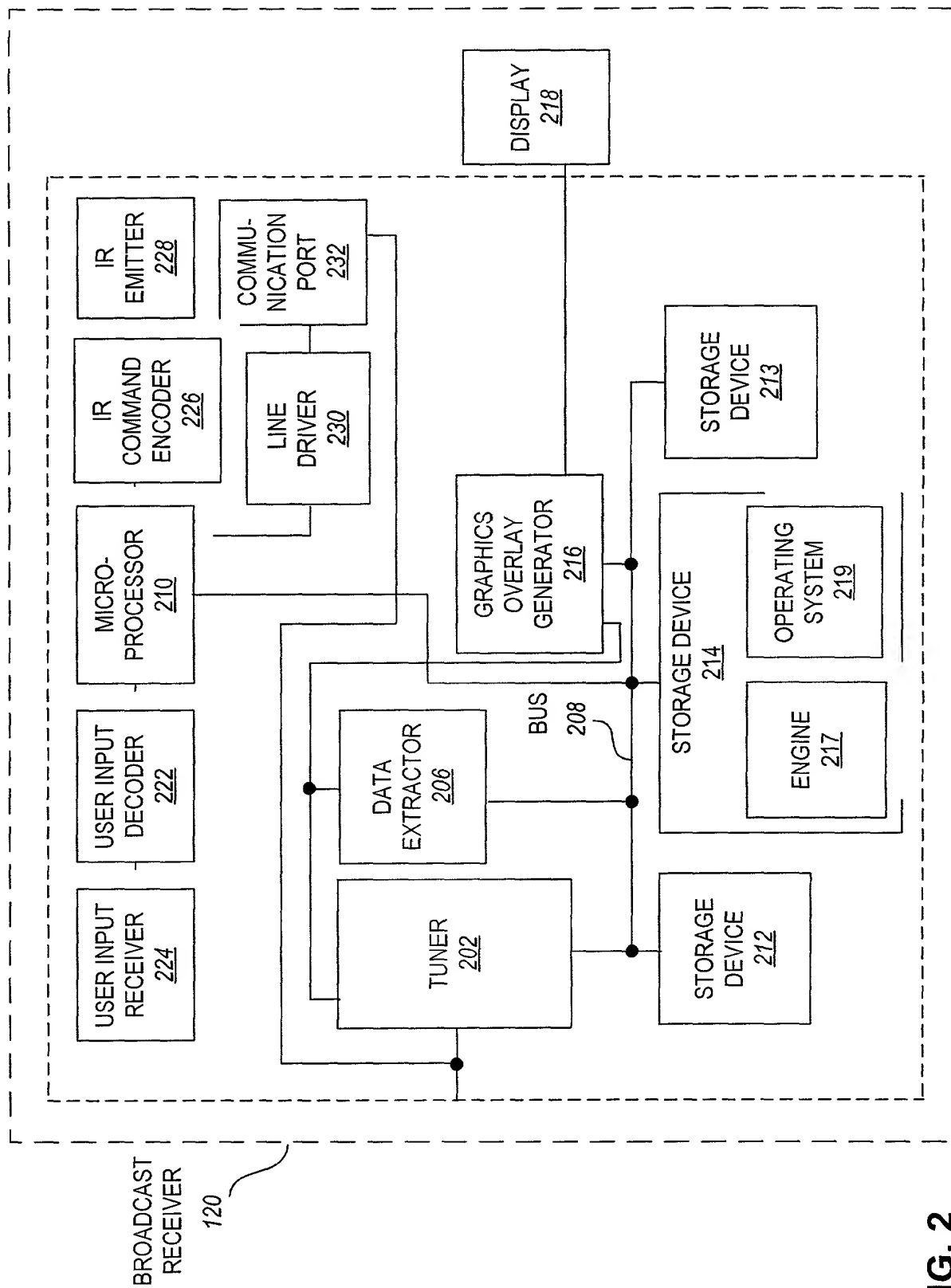
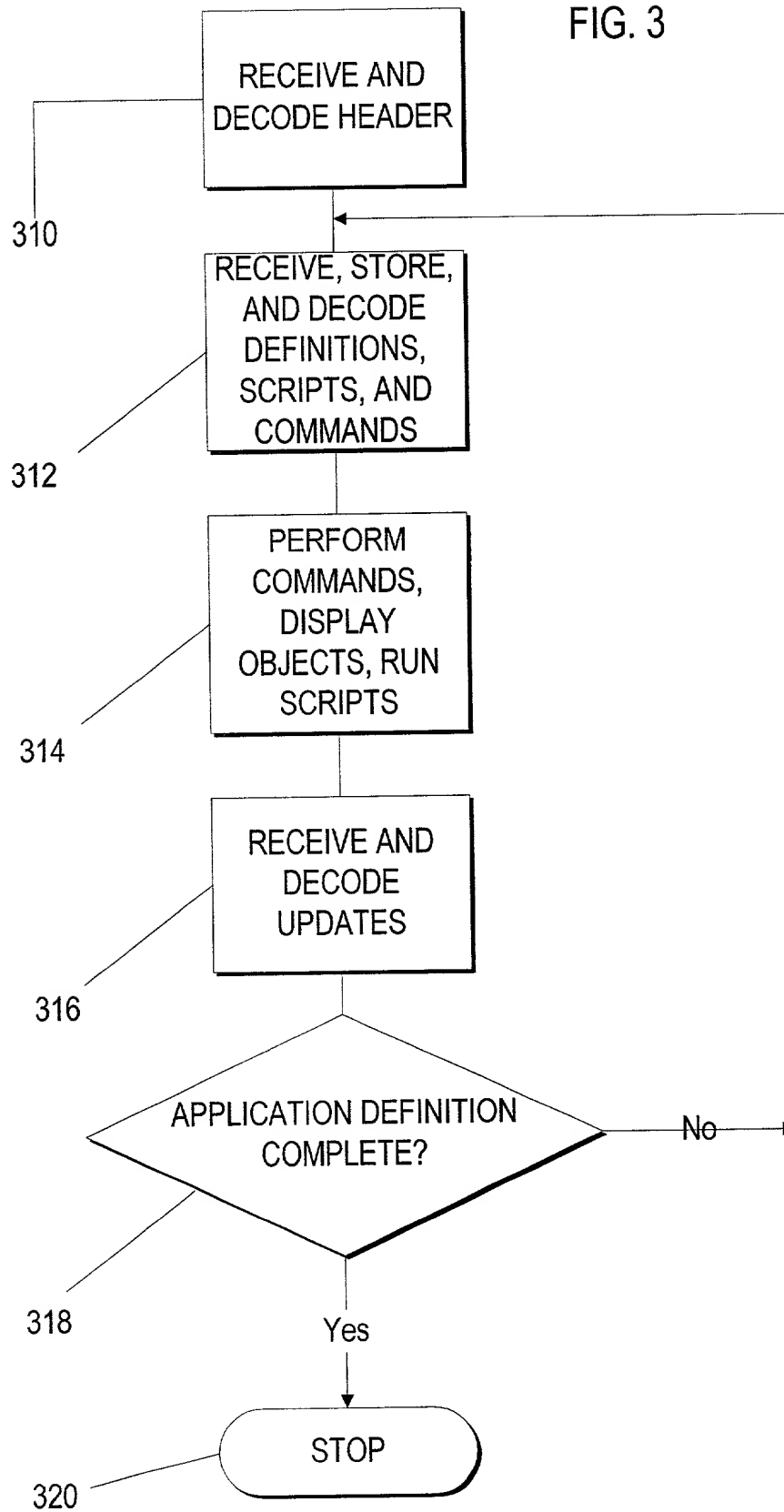


FIG. 2

FIG. 3



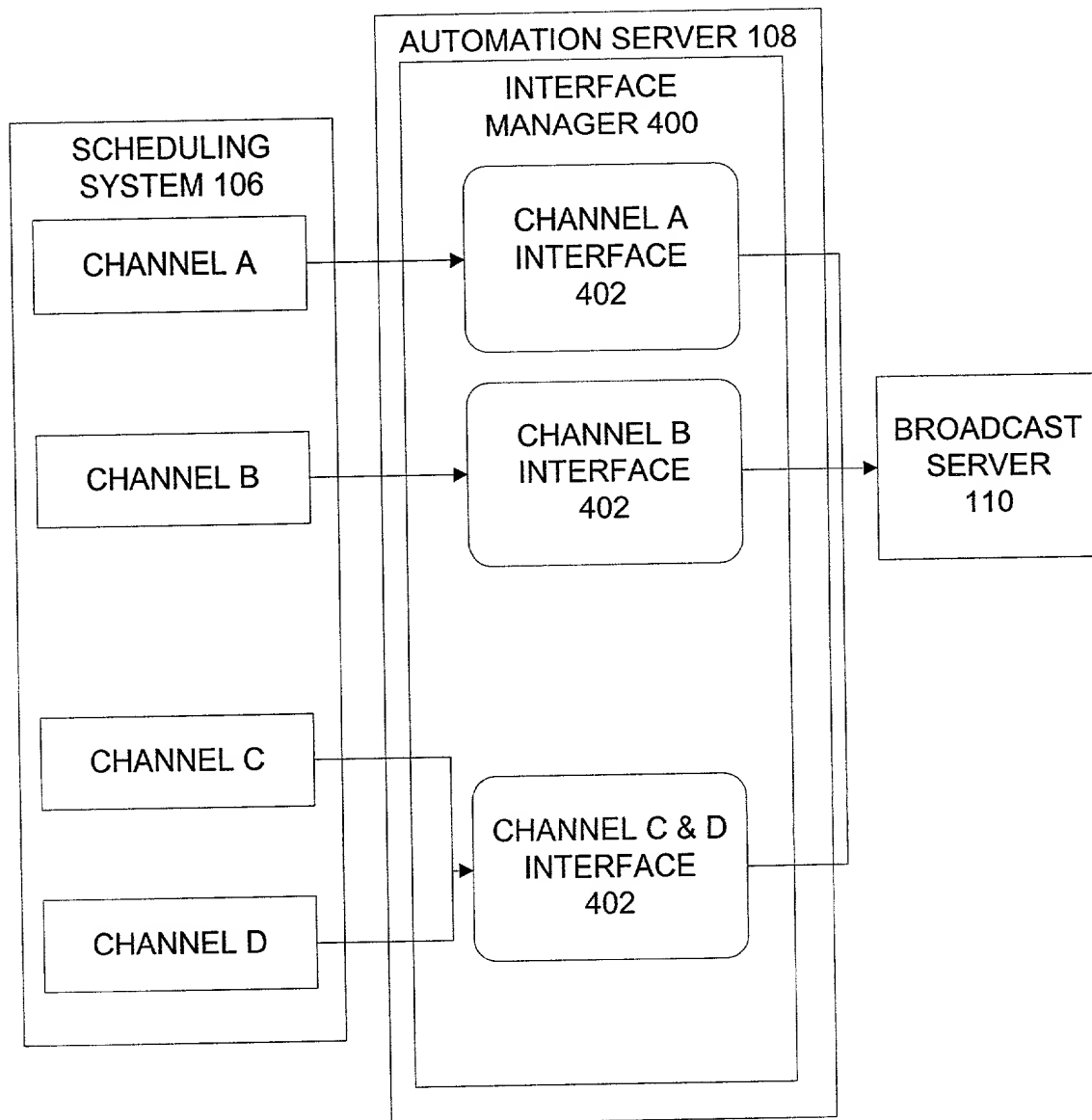


FIG. 4

FIG. 5 is a block diagram of a Channel Interface 402, according to one embodiment of the present invention. The Channel Interface 402 includes a Translator Module 502 and an Event Manager 504. The Translator Module 502 includes Mapping Logic 506, and a plurality of State Machines for Program #1 (510), State Machine for Program #2 (510), and State Machine for Program #N (510). The Event Manager 504 includes Mapping Logic 508, and a plurality of Event Managers #1 (512), Event Manager #2 (512), and Event Manager #M (512). The Channel Interface 402 is connected to a Scheduling System 106 and a Broadcast Server 110.

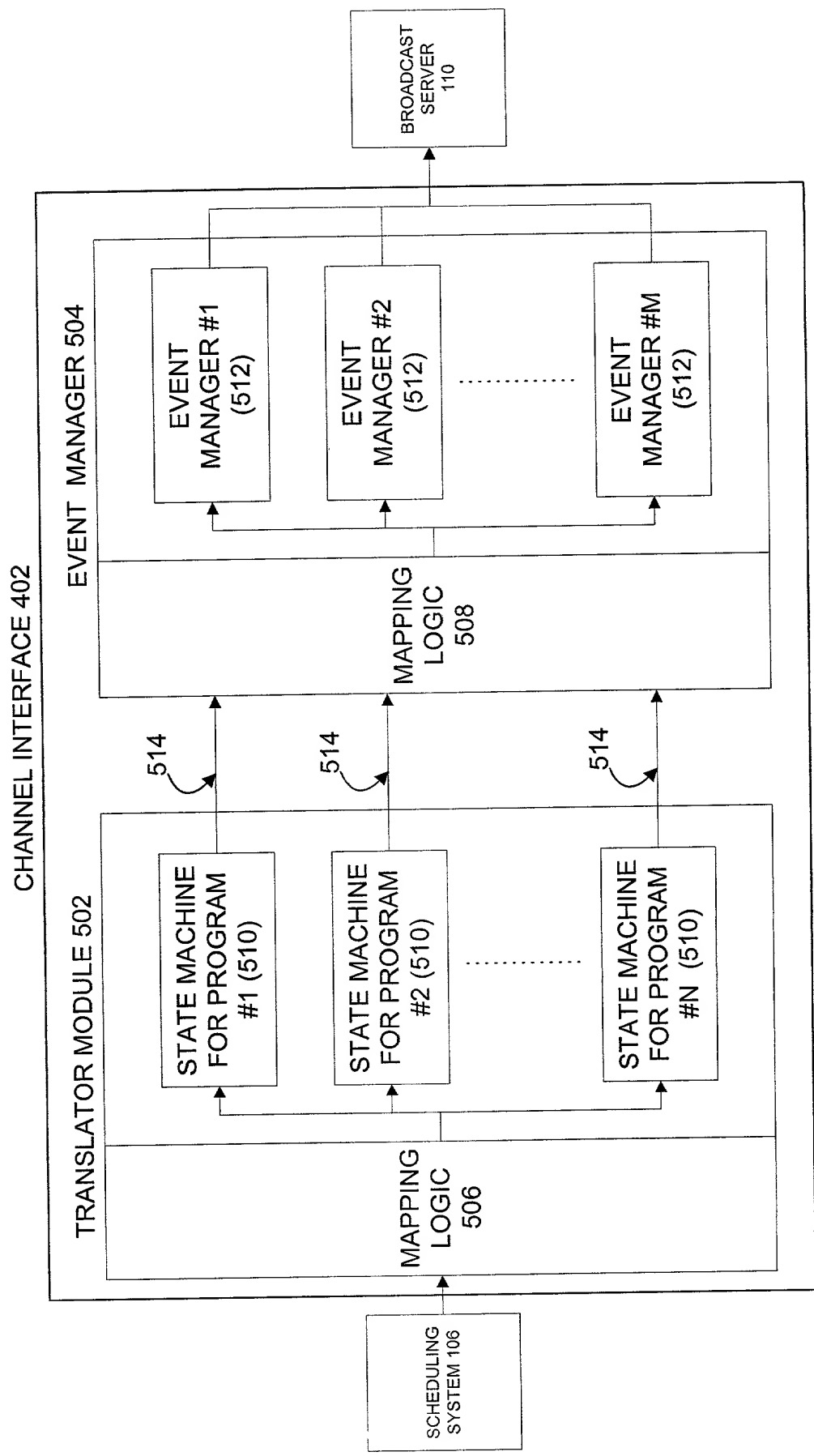


FIG. 5

Interface Manager Life Cycle Interaction Diagram

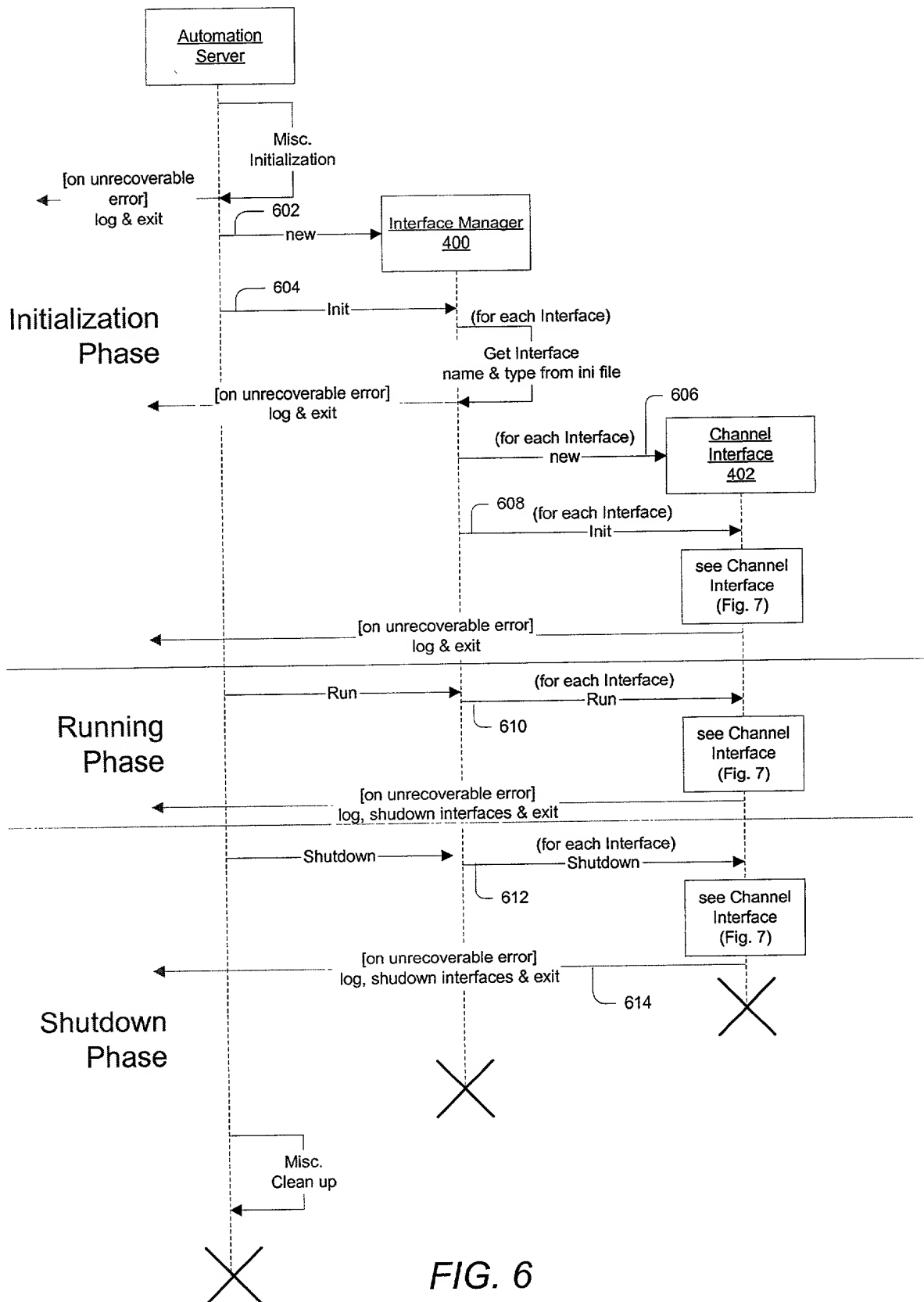


FIG. 6

Interface Life Cycle Interaction Diagram

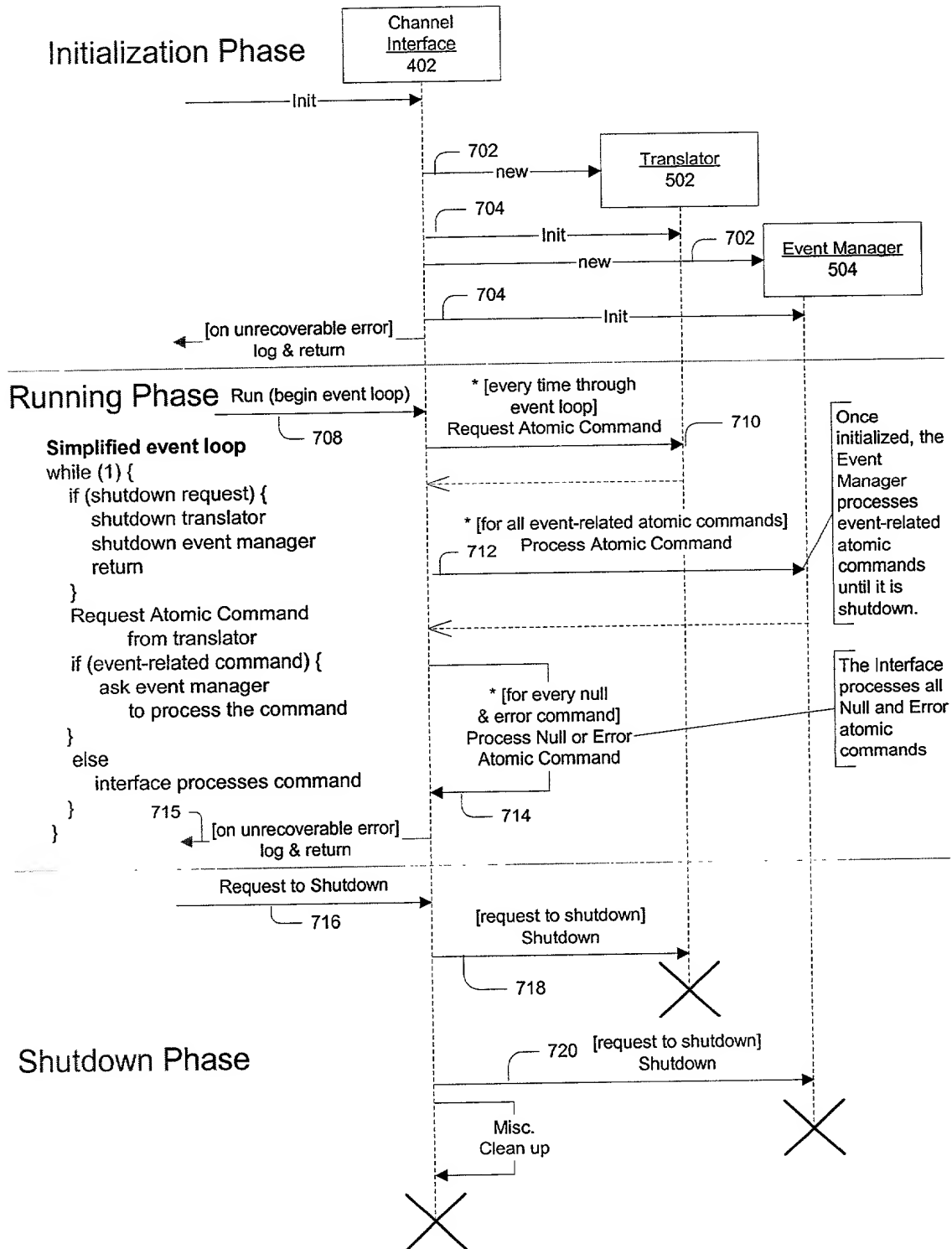
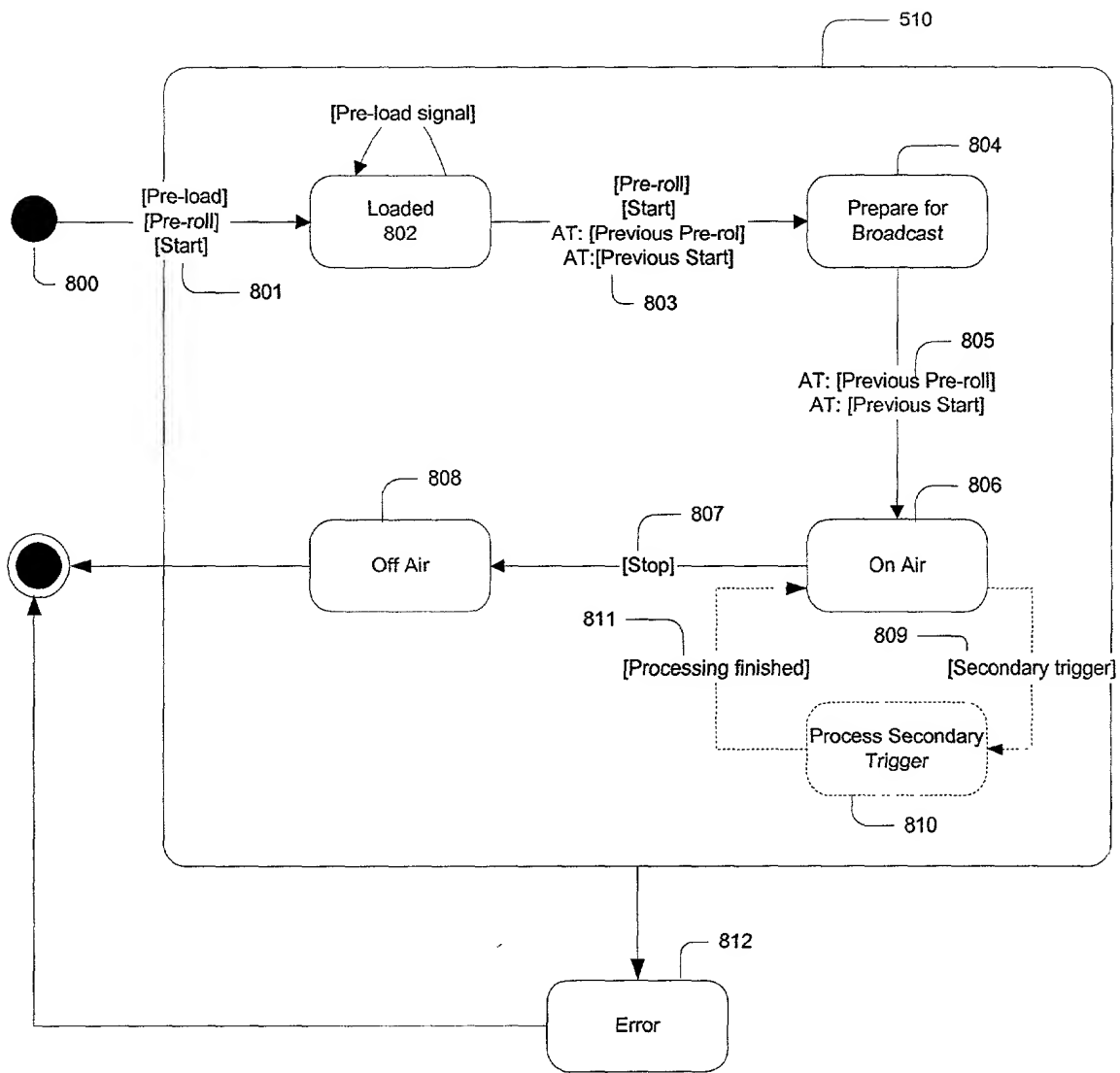


FIG. 7



AT: Automatic Transition on Previous Signal

FIG. 8

FIG. 9 is a flowchart illustrating a process for handling a scheduling system signal. The process begins with a "Scheduling System Signal" block (900), which leads to a decision diamond "Event Signal?" (902). If the answer is "NO", the process proceeds to a second decision diamond "Error Signal?" (904). If the answer to "Error Signal?" is "YES", the process proceeds to a block "Pass the Signal onto the Event Manager Module" (906). If the answer to "Error Signal?" is "NO", the process proceeds to a block "Ignore Signal" (910). If the answer to "Event Signal?" is "YES", the process proceeds to a third decision diamond "New Event?" (908). If the answer to "New Event?" is "YES", the process proceeds to a block "Create a Translator State Machine for the new Event" (912), which then leads to a block "Start the Translator State Machine for the new Event" (914). If the answer to "New Event?" is "NO", the process proceeds to a block "Pass Signal Onto the Translator State Machine for that Event" (916).

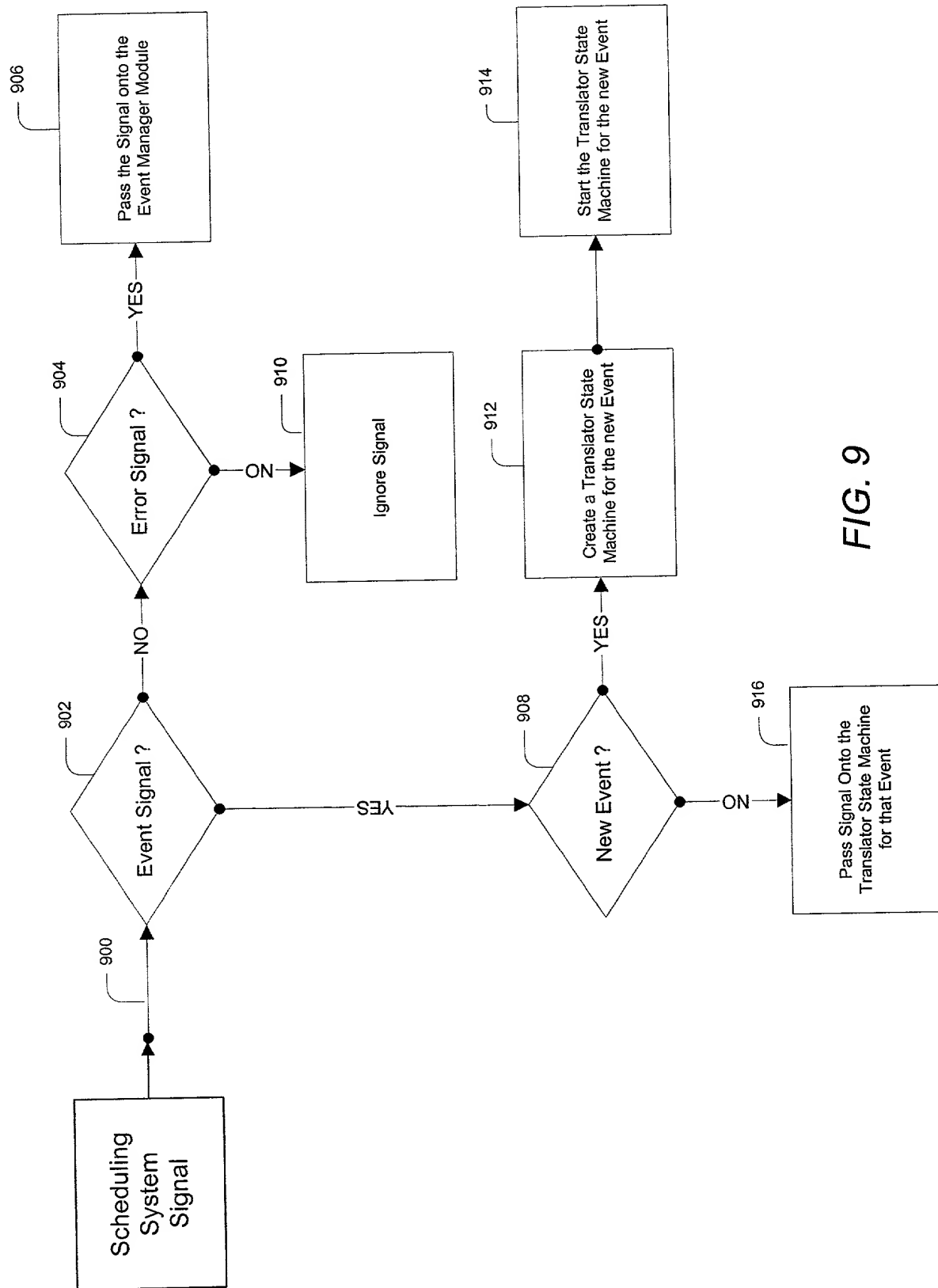


FIG. 9

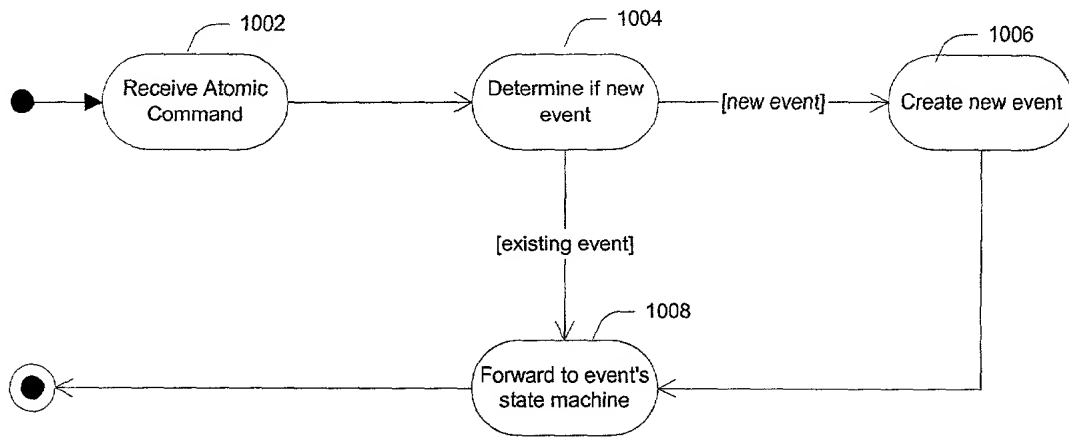


FIG. 10

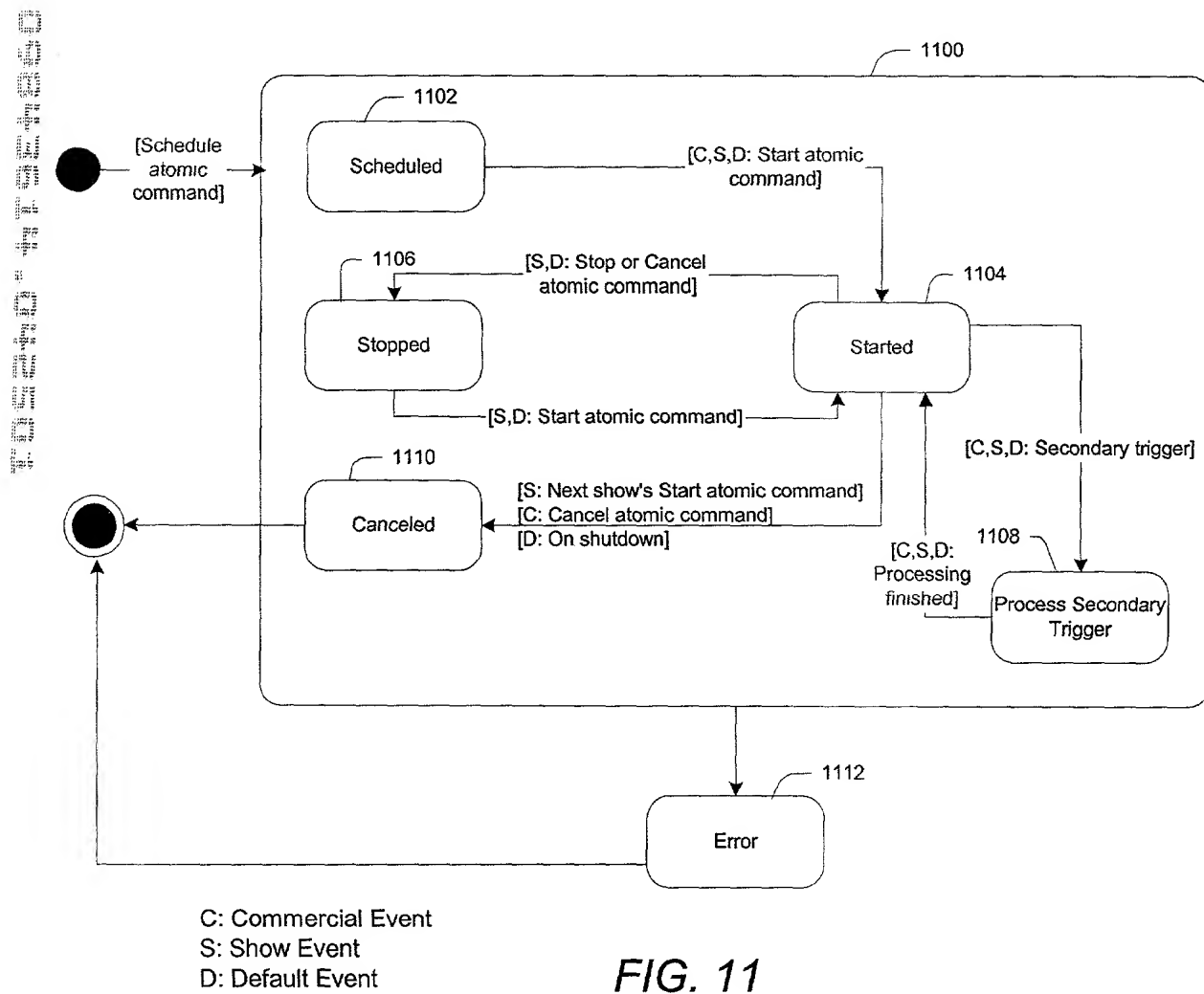


FIG. 11

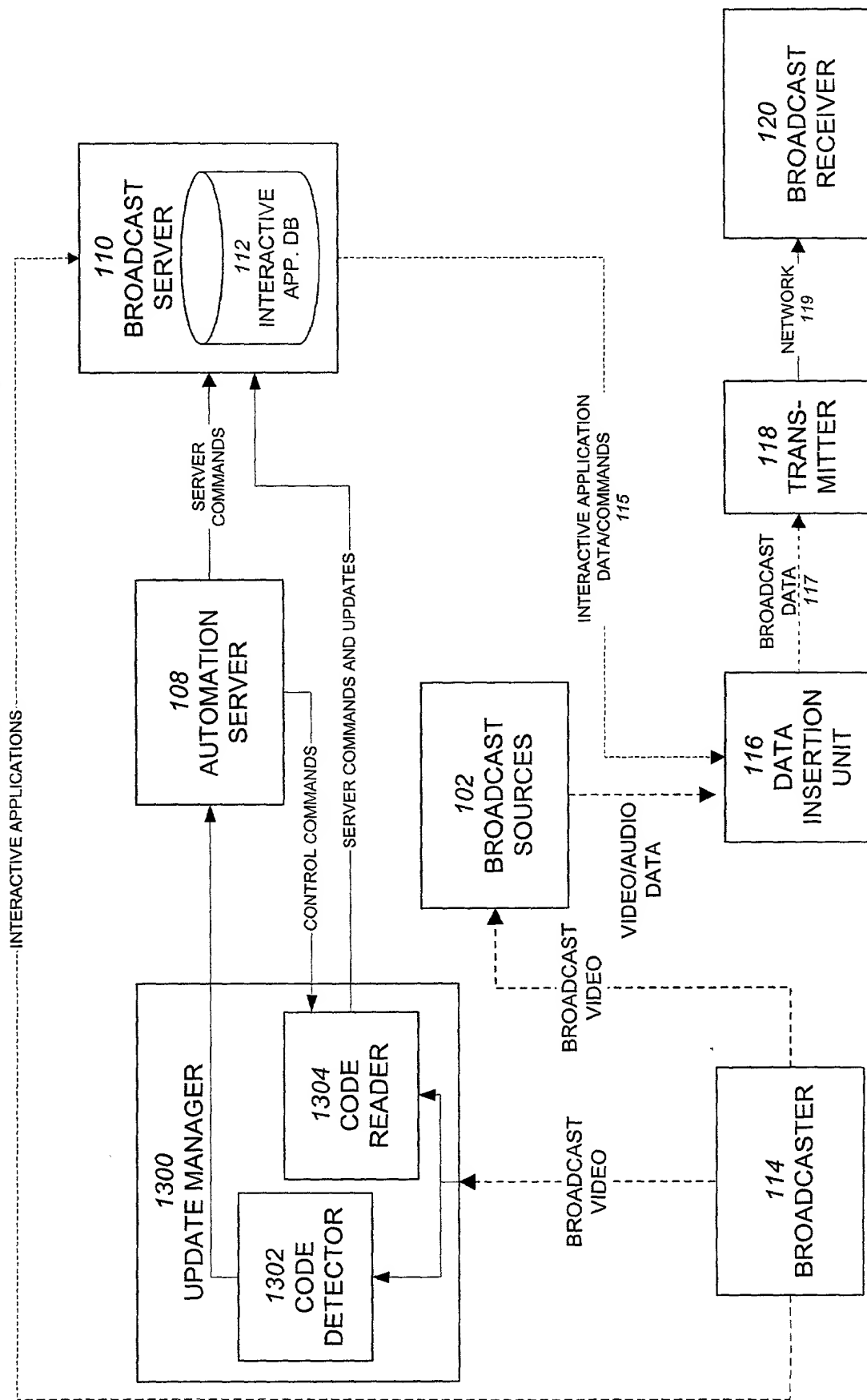


FIG. 13